Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 5, 2012

M. Boucadair
France Telecom
J. Touch
USC/ISI
P. Levis
France Telecom
R. Penno
Juniper Networks

September 2, 2011

Analysis of Solution Candidates to Reveal a Host Identifier in Shared Address Deployments

draft-boucadair-intarea-nat-reveal-analysis-04

Abstract

This document analyzes a set of solution candidates which have been proposed to mitigate some of the issues encountered when address sharing is used. In particular, this document focuses on means to reveal a host identifier when a Carrier Grade NAT (CGN) or application proxies are involved in the path. This host identifier must be unique to each host under the same shared IP address.

The ultimate goal is to assess the viability of proposed solutions and hopefully to make a recommendation on the more suitable solution(s).

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 5, 2012.

Table of Contents

1. Introduction

As reported in [RFC6269], several issues are encountered when an IP address is shared among several subscribers. Examples of such issues are listed below:

- Implicit identification (Section 13.2 of [RFC6269])
- SPAM (Section 13.3 of [RFC6269])
- Blacklisting a mis-behaving user (Section 13.1 of [RFC6269])
- Redirect users with infected machines to a dedicated portal (Section 5.1 of [RFC6269])

The sole use of the IPv4 address is not sufficient to uniquely distinguish a host. As a mitigation, it is tempting to investigate means which would help in disclosing an information to be used by the remote server as a means to uniquely disambiguate packets of hosts using the same IPv4 address.

The risk of not mitigating these issues are: OPEX increase for IP connectivity service providers (costs induced by calls to a hotline), revenue loss for content providers (loss of users audience), customers unsatisfaction (low quality of experience, service segregation, etc.).

1.1. Problem to Be Solved

Observation: Today, servers use the source IPv4 address as an identifier to treat some incoming connections differently. Tomorrow, due to the introduction of CGNs (e.g., NAT44 [I-D.ietf-behave-lsn-requirements], NAT64 [RFC6146]), that address will be shared. In particular, when a server receives packets from the same source address. Because this address is shared, the server does not know which host is the sending host.

Objective: The server should be able to sort out the packets by sending host.

Requirement: The server must have extra information than the source IP address to differentiate the sending host. We call HOST_ID this information.

For all solutions analyzed, we provide answers to the following questions:

What is the HOST_ID? It must be unique to each host under the same IP address. It does not need to be globally unique. Of course, the combination of the (public) IPv4 source address and the identifier (i.e., HOST_ID) ends up being relatively unique. As unique as today's 32-bit IPv4 addresses which, today, can change

when a host re-connects.

Where is the HOST_ID? (which protocol, which field): If the HOST_ID is put at the IP level, all packets will have to bear the identifier. If it is put at a higher connection-oriented level, the identifier is only needed once in the session establishment phase (for instance TCP three-way-handshake), then, all packets received in this session will be attributed to the HOST_ID designated during the session opening.

Who puts the HOST_ID? For almost all the analyzed solutions, the address sharing function injects the HOST_ID. When there are several address sharing functions in the data path, we describe to what extent the proposed solution is efficient. Another option to avoid potential performance degradation is to let the host inject its HOST_ID but the address sharing function will check its content (just like an IP anti-spoofing function).

What are the security considerations? Security considerations are common to all analyzed solutions (see Section 5). Privacy-related aspect are discussed in Section 1.2.

1.2. HOST_ID and Privacy

HOST_ID provides an additional information to uniquely disambiguate a host among those sharing the same IP address. Unlike URIs, HOST_ID does not leak user's identity information.

The HOST_ID does not reveal more privacy information than what the source IP address does in a non-shared address environment (see [I-D.morris-privacy-considerations]).

The volatility of the HOST_ID information is similar to the source IP address: a distinct HOST_ID may be used by the address sharing function when the host reboots or gets a new internal IP address. If the HOST_ID is persistent it may be used to track a host (similar to persistent IP addresses).

The trust on the information conveyed in the HOST_ID is likely to be the same as for current practices with the source IP address. In that sense, a HOST_ID can be spoofed as this is also the case for spoofing an IP address.

It is the responsibility of the remote server to rely or not on the content of the HOST_ID to enforce its policies and to log or not the content conveyed in the HOST_ID.

Enabling explicit identification means and adequate security suite is

more robust than relying on source IP address or HOST_ID. But tension may appear between strong privacy and usability (see Section 4.2 of [I-D.iab-privacy-workshop]).

1.3. IPv6 May Also Be Concerned

Issues similar to the ones described in Section 1.1 may be encountered also in an IPv6 environment (e.g., when the same /64 is used among several hosts).

1.4. Purpose and Scope

The purpose of this document is to analyze the solutions that have been proposed so far and to assess to what extent they solve the problem (see Section 1.1).

The purpose of this document is not to argue in favor of mandating the use of a HOST_ID but to document encountered issues, proposed solutions and their limitations.

Only IPv4-based solutions are analyzed in the following sections:

- define a new IP option (Section 3.1)
- define a new TCP option (Section 3.2)
- use the Identification field of IP header (denoted as IP-ID, Section 3.3)
- inject application headers (Section 3.4)
- enable Proxy Protocol ((Section 3.5))
- use of port set (Section 3.6)
- activate HIP (Section 3.7).

2. Recommendations

The following Table 1 summarizes the approaches analyzed in this document.

- "Success ratio" indicates the ratio of successful communications when the option is used. Provided figures are inspired from the results documented in [Options].
- "Deployable today" indicates if the solution can be generalized without any constraint on current architectures and practices.
- "Possible Perf Impact" indicates the level of expected performance degradation. The rationale behind the indicated potential performance degradation is whether the injection requires some treatment at the IP level or not.

o "OS TCP/IP Modif" indicates whether a modification of the OS TCP/IP stack is required at the server side.

	IP Option	TCP Option	IP-ID	HTTP Header (XFF)	Proxy Protocol	Port Set	HIP
UDP	Yes	No	Yes	No	No	Yes	
TCP	Yes	Yes	Yes	No	Yes	Yes	
HTTP	Yes	Yes	Yes	Yes	Yes	Yes	
Encrypted Traffic	Yes	Yes	Yes	No	Yes	Yes	
Success Ratio	30%	99%	100%	100%	Low	100%	Low
Possible Perf Impact	High	Med to High	Low to Med	Med to High	High	No	N/A
OS TCP/IP Modif	Yes	Yes	Yes	No	No	No	
Deployable Today	Yes	Yes	Yes	Yes	No	Yes	No
Notes			(1)	(2)		(1) (3)	(4) (5)

Table 1: Summary of analyzed solutions.

Notes for the above table:
(1) Requires mechanism to advertise NAT is participating in this scheme (e.g., DNS PTR record)
(2) This solution is widely deployed
(3) When the port set is not advertised, the solution is less efficient for third-party services.
(4) Requires the client and the server to be HIP-compliant and HIP infrastructure to be deployed.

(5) If the client and the server are HIP-enabled, the address sharing function does not need to insert a host-hint. If the client is not HIP-enabled, designing the device that performs address sharing to act as a UDP/TCP-HIP relay is not viable.

According to the above table and the analysis elaborated in Section 3:

- IP Option, IP-ID and Proxy Protocol proposals are broken;
- HIP is not largely deployed;
- The use of Port Set may contradict the port randomization [RFC6056] requirement identified in [RFC6269]. This solution can be used by a service provider for the delivery of its own service offerings relying on implicit identification.
- XFF is de facto standard deployed and supported in operational networks (e.g., HTTP Severs, Load-Balancers, etc.).
- From an application standpoint, the TCP Option is superior to XFF since it is not restricted to HTTP. Nevertheless XFF is compatible with the presence of address sharing and load-balancers in the communication path. To provide a similar functionality, the TCP Option may be extended to allow conveying a list of IP addresses to not loose the source IP address in the presence of load-balancers. Note that TCP Option requires the modification of the OS TCP/IP stack of remote servers; which can be seen as a blocking point.

As a conclusion of this analysis, the following recommendation is made:

[Hopefully to be completed]

3. Solutions Analysis

3.1. Define an IP Option

3.1.1. Description

This proposal aims to define an IP option [RFC0791] to convey a "host identifier". This identifier can be inserted by the address sharing function to uniquely distinguish a host among those sharing the same IP address. The option can convey an IPv4 address, the prefix part of an IPv6 address, etc.

Another way for using IP option has been described in Section 4.6 of [RFC3022].

3.1.2. Analysis

Unlike the solution presented in Section 3.2, this proposal can apply for any transport protocol. Nevertheless, it is widely known that routers (and other middle boxes) filter IP options. IP packets with IP options can be dropped by some IP nodes. Previous studies demonstrated that "IP Options are not an option" (Refer to [Not_An_Option], [Options]).

As a conclusion, using an IP option to convey a host-hint is not viable.

3.2. Define a TCP Option

3.2.1. Description

This proposal [I-D.wing-nat-reveal-option] defines a new TCP option called USER_HINT. This option encloses the TCP client's identifier (e.g., the lower 16 bits of their IPv4 address, their VLAN ID, VRF ID, subscriber ID). The address sharing device inserts this TCP option to the TCP SYN packet.

3.2.2. Analysis

The risk related to handling a new TCP option is low as measured in [Options].

[I-D.wing-nat-reveal-option] discusses the interference with other TCP options.

Using a new TCP option to convey the host-hint does not require any modification to the applications but it is applicable only for TCP-based applications. Applications relying on other transport protocols are therefore left unsolved.

Some downsides have been raised against defining a TCP option to reveal a host identity:

- Conveying an IP address in a TCP option may be seen as a violation of OSI layers but since IP addresses are already used for the checksum computation, this is not seen as a blocking point. Moreover, Updated version of [I-D.wing-nat-reveal-option] does not allow anymore to convey an IP address (the HOST_ID is encoded in 16bits).

- TCP option space is limited, and might be consumed by the TCP client. Earlier versions of [I-D.wing-nat-reveal-option] discuss two approaches to sending the HOST_ID: sending the HOST_ID in the TCP SYN (which consumes more bytes in the TCP header of the TCP SYN) and sending the HOST_ID in a TCP ACK (which consumes only two bytes in the TCP SYN). Content providers may find it more desirable to receive the HOST_ID in the TCP SYN, as that more closely preserves the host hint received in the source IP address as per current practices. It is more complicated to implement sending the HOST_ID in a TCP ACK, as it can introduce MTU issues if the ACK packet also contains TCP data, or a TCP segment is lost. The latest specification of the HOST_ID TCP Option, documented at [I-D.wing-nat-reveal-option], allows only to enclose the HOST_ID in the TCP SYN packet.

- When there are several NATs in the path, the original HOST_ID may be lost. In such case, the procedure may not be efficient.

- Interference with current usages such as X-Forwarded-For (see Section 3.4) should be elaborated to specify the behavior of servers when both options are used; in particular specify which information to use: the content of the TCP option or what is conveyed in the application headers.

- When load-balancers or proxies are in the path, this option does not allow to preserve the original source IP address and source. Preserving such information is required for logging purposes for instance.

3.3. Use the Identification Field of IP Header (IP-ID)

3.3.1. Description

IP-ID (Identification field of IP header) can be used to insert an information which uniquely distinguishes a host among those sharing the same IPv4 address. An address sharing function can re-write the IP-ID field to insert a value unique to the host (16 bits are sufficient to uniquely disambiguate hosts sharing the same IP address). Note that this field is not altered by some NATs; hence some side effects such as counting hosts behind a NAT as reported in [Count].

A variant of this approach relies upon the format of certain packets, such as TCP SYN, where the IP-ID can be modified to contain a 16 bit host-hint. Address sharing devices performing this function would require to indicate they are performing this function out of band, possibly using a special DNS record.

3.3.2. Analysis

This usage is not compliant with what is recommended in [I-D.ietf-intarea-ipv4-id-update].

3.4. Inject Application Headers

3.4.1. Description

Another option is to not require any change at the transport nor the IP levels but to convey at the application payload the required information which will be used to disambiguate hosts. This format and the related semantics depend on its application (e.g., HTTP, SIP, SMTP, etc.).

For HTTP, the X-Forwarded-For (XFF) header can be used to display the original IP address when an address sharing device is involved. Service Providers operating address sharing devices can enable the feature of injecting the XFF header which will enclose the original IPv4 address or the IPv6 prefix part. The address sharing device has to strip all included XFF headers before injecting their own. Servers may rely on the contents of this field to enforce some policies such as blacklisting misbehaving users. Note that XFF can also be logged by some servers (this is for instance supported by Apache).

3.4.2. Analysis

Not all applications impacted by the address sharing can support the ability to disclose the original IP address. Only a subset of protocols (e.g., HTTP) can rely on this solution.

For the HTTP case, to prevent users injecting invalid host-hints, an initiative has been launched to maintain a list of trusted ISPs using XFF: See for example the list available at: [Trusted_ISPs] of trusted ISPs as maintained by Wikipedia. If an address sharing device is on the trusted XFF ISPs list, users editing Wikipedia located behind the address sharing device will appear to be editing from their "original" IP address and not from the NATed IP address. If an offending activity is detected, individual hosts can be blacklisted instead of all hosts sharing the same IP address.

XFF header injection is a common practice of load balancers. When a load balancer is in the path, the original content of any included XFF header should not be stripped. Otherwise the information about the "origin" IP address will be lost.

When several address sharing devices are crossed, XFF header can

convey the list of IP addresses. The origin HOST_ID can be exposed to the target server.

XFF also introduces some implementation complexity if the HTTP packet is at or close to the MTU size.

It has been reported that some "poor" implementation may encounter some parsing issues when injecting XFF header.

For encrypted HTTP traffic, injecting XFF header may be broken.

3.5. PROXY Protocol

3.5.1. Description

The solution, referred to as Proxy Protocol [Proxy], does not require any application-specific knowledge. The rationale behind this solution is to prepend each connection with a line reporting the characteristics of the other side's connection as shown in the example below (excerpt from [Proxy]):

```
PROXY TCP4 198.51.100.1 198.51.100.11 56324 443\r\n
```

Upon receipt of a message conveying this line, the server removes the line. The line is parsed to retrieve the transported protocol. The content of this line is recorded in logs and used to enforce policies.

3.5.2. Analysis

This solution can be deployed in a controlled environment but it can not be deployed to all access services available in the Internet. If the remote server does not support the Proxy Protocol, the session will fail. Other complications will raise due to the presence of firewalls for instance.

As a consequence, this solution is broken and can not be recommended.

3.6. Enforce a Source-based Selection Algorithm at the Server Side (Port Set)

3.6.1. Description

This solution proposal does not require any action from the address sharing function to disclose a host identifier. Instead of assuming all the ports are associated with the same host, a random-based algorithm (or any port selection method) is run to generate the set of ports (including the source port of the received packet). The

length of the ports set to be generated by the server may be configurable (e.g., 8, 32, 64, 512, 1024, etc.). Instead of a random-based scheme, the server can use contiguous port ranges to form the port sets.

The server may reduce (or enlarge) the width of the ports set of the misbehaving action is (not) mitigated.

A variant of this proposal is to announce by off-line means the port set assignment policy of an operator. This announcement is not required for the delivery of internal services (i.e., offered by the service provider deploying the address sharing function) relying on implicit identification.

3.6.2. Analysis

In nominal mode, no coordination is required between the address sharing function and the server side but the efficiency of the method depends on the port set selection algorithm.

The method is more efficient if the provider that operates the address sharing device advertises its port assignment policy but this may contradicts the port randomization as identified in [RFC6269].

The method is deterministic for the delivery of services offered by the service provider offering also the IP connectivity service.

3.7. Host Identity Protocol (HIP)

3.7.1. Description

[RFC5201] specifies an architecture which introduces a new namespace to convey an identity information.

3.7.2. Analysis

This solution requires both the client and the server to support HIP [RFC5201]. Additional architectural considerations are to be taken into account such as the key exchanges, etc.

If the address sharing function is required to act as a UDP/TCP-HIP relay, this is not a viable option.

4. IANA Considerations

This document does not require any action from IANA.

5. Security Considerations

The same security concerns apply for the injection of an IP option, TCP option and application-related content (e.g., XFF) by the address sharing device. If the server trusts the content of the HOST_ID field, a third party user can be impacted by a misbehaving user to reveal a "faked" original IP address.

6. Acknowledgments

Many thanks to D. Wing and C. Jacquenet for their review, comments and inputs.

Thanks also to P. McCann, T. Tsou, Z. Dong, B. Briscoe, T. Taylor, M. Blanchet, D. Wing and A. Yourtchenko for the discussions in Prague.

Some of the issues related to defining a new TCP option have been raised by L. Eggert.

7. References

7.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.

7.2. Informative References

[Count] "A technique for counting NATted hosts", <http://www.cs.columbia.edu/~smb/papers/fnat.pdf>.

[I-D.iab-privacy-workshop] Cooper, A., "Report from the Internet Privacy Workshop", draft-iab-privacy-workshop-00 (work in progress), June 2011.

[I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NAT (CGN)", draft-ietf-behave-lsn-requirements-03 (work in progress), August 2011.

[I-D.ietf-intarea-ipv4-id-update]
Touch, J., "Updated Specification of the IPv4 ID Field", draft-ietf-intarea-ipv4-id-update-02 (work in progress), March 2011.

[I-D.morris-privacy-considerations]
Aboba, B., Morris, J., Peterson, J., and H. Tschofenig, "Privacy Considerations for Internet Protocols", draft-morris-privacy-considerations-03 (work in progress), March 2011.

[I-D.wing-nat-reveal-option]
Yourtchenko, A. and D. Wing, "Revealing hosts sharing an IP address using TCP option", draft-wing-nat-reveal-option-02 (work in progress), June 2011.

[Not_An_Option]
R. Fonseca, G. Porter, R. Katz, S. Shenker, and I. Stoica,, "IP options are not an option", 2005, <http://www.eecs.berkeley.edu/Pubs/TechRpts/2005/EECS-2005-24.html>.

[Options] Alberto Medina, Mark Allman, Sally Floyd, "Measuring Interactions Between Transport Protocols and Middleboxes", 2005, <http://conferences.sigcomm.org/imc/2004/papers/p336-medina.pdf>.

[Proxy] Tarreau, W., "The PROXY protocol", November 2010, <http://haproxy.1wt.eu/download/1.5/doc/proxy-protocol.txt>.

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[Trusted_ISPs]
"Trusted XFF list", <http://meta.wikimedia.org/wiki/XFF_project#Trusted_XFF_list>.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

Joe Touch
USC/ISI

Email: touch@isi.edu

Pierre Levis
France Telecom
Caen, 14000
France

Email: pierre.levis@orange-ftgroup.com

Reinaldo Penno
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, California 94089
USA

Email: rpenno@juniper.net

V6OPS
Internet-Draft
Intended status: Informational
Expires: April 15, 2012

B. Carpenter
Univ. of Auckland
S. Jiang
Huawei Technologies Co., Ltd

October 13, 2011

Using the IPv6 Flow Label for Server Load Balancing
draft-carpenter-v6ops-label-balance-00

Abstract

This document describes how the IPv6 flow label can be used in support of layer 3/4 load balancing for large server farms.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 15, 2012.

Table of Contents

1. Introduction

The IPv6 flow label has been redefined [I-D.ietf-6man-flow-3697bis] and its use for load balancing in multipath routing has been specified [I-D.ietf-6man-flow-ecmp]. Another scenario in which the flow label could be used is in load balancing for large server farms. This document starts with a brief introduction to load balancing techniques and then describes how the flow label can be used to enhance layer 3/4 flow balancers in particular.

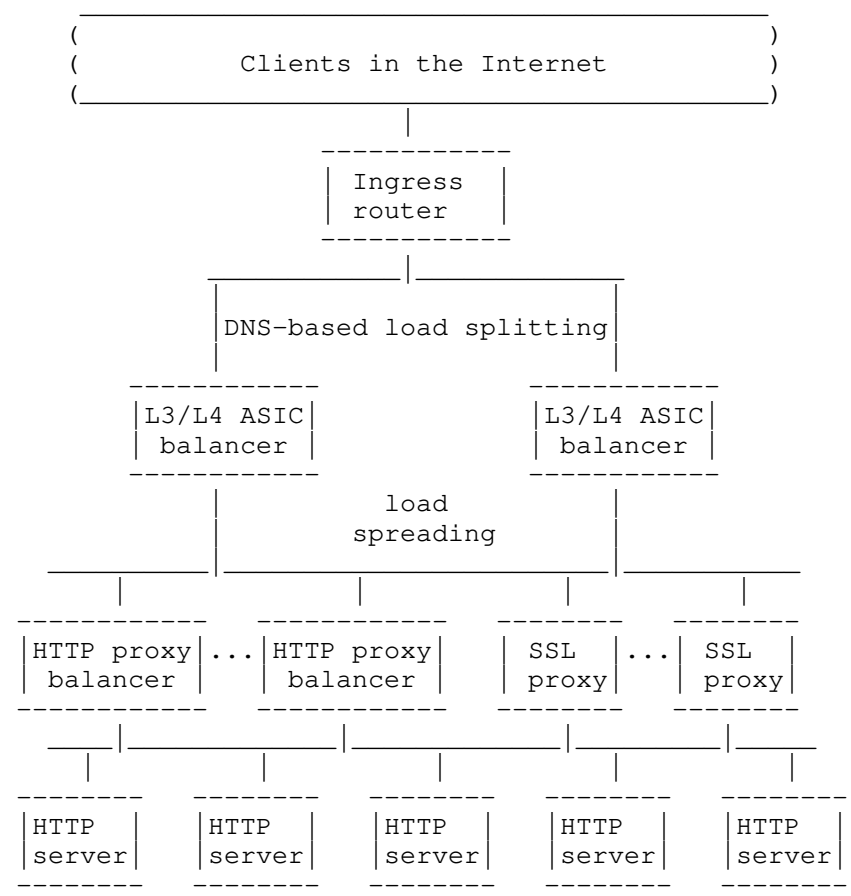
Load balancing for server farms is achieved by a variety of methods, often used in combination [Tarreau]. The flow label is not relevant to all of them. Also, the actual load balancing algorithm (the choice of server for a new client session) is irrelevant to this discussion.

- The simplest method is simply using the DNS to return different server addresses for a single name such as www.example.com to different users. Typically this is done by rotating the order in which different addresses are listed by the relevant authoritative DNS server, assuming that the client will pick the first one. The flow label can have no impact on this method and it is not discussed further.
- Another method, for HTTP servers, is to operate a layer 7 reverse proxy in front of the server farm. The reverse proxy will present a single IP address to the world, communicated to clients by a single AAAA record. For each new client session (an incoming TCP connection and HTTP request), it will pick a particular server and proxy the session to it. Hopefully the act of proxying will be cheap compared to the act of serving the required content. The proxy must retain TCP state and proxy state for the duration of the session. This TCP state could, potentially, include the incoming flow label value.
- A component of some load balancing systems is an SSL reverse proxy farm. The individual SSL proxies handle all cryptographic aspects and exchange raw HTTP with the actual servers. Thus, from the load balancing point of view, this really looks just like a server farm, except that it's specialised for HTTPS. Each proxy will retain SSL and TCP and maybe HTTP state for the duration of the session, and the TCP state could potentially include the flow label.
- Finally the "front end" of many load balancing systems is a layer 3/4 load balancer. In this case, it is the layer 3/4 load balancer whose IP address is published as the primary AAAA record for the service. All client sessions will pass through this device. According to the precise scenario, it will spread new sessions across the actual application servers, across an SSL proxy farm, or across a set of layer 7 proxies. In all cases, the

layer 3/4 load balancer has to recognize incoming packets as belonging to new or existing client sessions, and choose the target server or proxy so as to ensure persistence. 'Persistence' is defined as guaranteeing that a given session will run to completion on a single server. The layer 3/4 load balancer, whatever method it uses for forwarding the session, is certain to inspect the source address and the protocol and port numbers in each incoming packet. At the same time, it could inspect and make use of the flow label.

Layer 3/4 load balancers use various techniques to actually reach their target server.
- All servers are configured with the same IP address, they are all on the same LAN, and the load balancer sends directly to their individual MAC addresses.
- Each server has its own IP address, and the balancer uses an IP-in-IP tunnel to reach it.
- Each server has its own IP address, and the balancer performs NAPT (address and port translation).

The following diagram, inspired by [Tarreau], shows a maximum layout.



From the previous paragraphs, we can identify several points in this diagram where the flow label may be relevant:

1. L3/L4 load balancers.
2. SSL proxies.
3. HTTP proxies.

2. Role of the Flow Label

The IPv6 flow label is included in every IPv6 header [RFC2460] and it is defined in [I-D.ietf-6man-flow-3697bis]. According to this definition, it should be set to a constant value for a given traffic flow (such as an HTTP connection), but until the standard is widely implemented it will often be set to the default value of zero. Any device that has access to the IPv6 header has access to the flow

label, and it is at a fixed position in every IPv6 packet. In contrast, transport layer information, such as the port numbers, is not always in a fixed position, since it follows any IPv6 extension headers that may be present. Therefore, within the lifetime of a given transport layer connection, the flow label can be a more convenient "handle" than the port number for identifying that particular connection.

According to [I-D.ietf-6man-flow-3697bis], source hosts should set the flow label, but if they do not (i.e. its value is zero), forwarding nodes may do so instead. In both cases, the flow label value must be constant for a given transport session, normally identified by the IPv6 and Transport header 5-tuple. The flow label should be calculated by a stateless algorithm. The value should form part of a statistically uniform distribution, making it suitable as part of a hash function used for load distribution. Because of using a stateless algorithm to calculate the label, there is a very low (but non-zero) probability that two simultaneous flows from the same source to the same destination have the same flow label value despite having different transport protocol port numbers.

The suggested model for using the flow label in a load balancing mechanism is as follows.

- It is clearly better if the original source, e.g. an HTTP client, sets the flow label. However, if the flow label of an incoming packet is zero, the ingress router at the server site should implement the stateless mechanism in Section 3 of [I-D.ietf-6man-flow-3697bis] to set the flow label value to an appropriate value. This relieves the subsequent load balancers of the need to fully analyse the IPv6 and Transport header 5-tuple.
- The L3/L4 load balancers use the 2-tuple {source address, flow label} as the session key for whatever load distribution algorithm they support, instead of searching for the transport port number later in the header. This means they can ignore all IPv6 extension headers, which should simplify their design and lead to a performance benefit.
- The SSL proxies may do the same. However, since they have to process the transport layer in any case, this might not lead to any performance benefit.
- The HTTP proxies may do the same. However, since they have to process the transport and application layers in any case, this might not lead to any performance benefit.

Note that in the unlikely event of two simultaneous flows from the same source having the same flow label value, the two flows would end up assigned to the same server, where they would be distinguished as normal by their port numbers. Since this would be a statistically

rare event, it would not damage the overall load balancing effect.

3. Security Considerations

Security aspects of the flow label are discussed in [I-D.ietf-6man-flow-3697bis]. As noted there, a malicious source or man-in-the-middle could disturb load balancing by manipulating flow labels.

Specifically, [I-D.ietf-6man-flow-3697bis] states that "stateless classifiers should not use the flow label alone to control load distribution, and stateful classifiers should include explicit methods to detect and ignore suspect flow label values." The former point is answered by also using the source address. The latter point is more complex. If the risk is considered serious, the ingress router mentioned above should verify incoming flows with non-zero flow label values. If a flow from a given source address and port number does not have a constant flow label value, it is suspect and should be dropped.

4. IANA Considerations

This document requests no action by IANA.

5. Acknowledgements

Valuable comments and contributions were made by

This document was produced using the xml2rfc tool [RFC2629].

6. Change log [RFC Editor: Please remove]

draft-carpenter-v6ops-label-balance-00: original version, 2011-10-13.

7. References

7.1. Normative References

[I-D.ietf-6man-flow-3697bis]
Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", draft-ietf-6man-flow-3697bis-07 (work in progress), July 2011.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

7.2. Informative References

[I-D.ietf-6man-flow-ecmp] Carpenter, B. and S. Amante, "Using the IPv6 flow label for equal cost multipath routing and link aggregation in tunnels", draft-ietf-6man-flow-ecmp-05 (work in progress), July 2011.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[Tarreau] Tarreau, W., "Making applications scalable with load balancing", 2006, <http://1wt.eu/articles/2006_lb/>.

Authors' Addresses

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland, 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing 100095
P.R. China

Email: jiangsheng@huawei.com

INTAREA WG
Internet-Draft
Updates: 4861 (if approved)
Intended status: Standards Track
Expires: May 3, 2012

S. Chakrabarti
Ericsson
E. Nordmark
Cisco Systems
M. Wasserman
Painless Security

October 31, 2011

Energy Aware IPv6 Neighbor Discovery Optimizations
draft-chakrabarti-nordmark-energy-aware-nd-01

Abstract

IPv6 Neighbor Discovery (RFC 4861) protocol has been designed for neighbor's address resolution, unreachability detection, address autoconfiguration, router advertisement and solicitation. With the progress of Internet adoption on various industries including home, wireless and machine-to-machine communications, there is a desire for optimizing legacy IPv6 Neighbor Discovery protocol for energy-efficient networks and nodes. Research indicates that often networked- nodes require more energy than stand-alone nodes because a node's energy usage depends on network messages it receives and responds. While reducing energy consumption is essential for battery operated nodes in some machines, saving energy actually a cost factor in business in general as the explosion of more device usage is leading to usage of more servers and network infrastructure in all sectors of the society and business. This document describes a method of optimizations by reducing periodic multicast messages, frequent Neighbor Solicitation messages and discusses interoperability with legacy IPv6 nodes. This document also addresses the ND denial of service issues by introducing node Registration procedure.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2012.

Table of Contents

1. Introduction

IPv6 ND [ND] is based on multicast signaling messages on the local link in order to avoid broadcast messages. Following power-on and initialization of the network in IPv6 Ethernet networks, a node joins the solicited-node multicast address on the interface and then performs duplicate address detection (DAD) for the acquired link-local address by sending a solicited-node multicast message to the link. After that it sends multicast router solicitation (RS) messages to the all-router address to solicit router advertisements. Once the host receives a valid router advertisement (RA) with the "A" flag, it autoconfigures the IPv6 address with the advertised prefix in the router advertisement (RA). Besides this, the IPv6 routers usually send router advertisements periodically on the network. RAs are sent to the all-node multicast address. Nodes send Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages to resolve the IPv6 address of the destination on the link. These NS/NA messages are also often multicast messages and it is assumed that the node is on the same link and relies on the fact that the destination node is always powered and generally available.

The periodic RA messages in IPv6 ND [ND], and NS/NA messages require all IPv6 nodes in the link to be in listening mode even when they are in idle cycle. It requires energy for the sleepy nodes which may otherwise be sleeping during the idle period. Non-sleepy nodes also save energy if instead of continuous listening, they actually pro-actively synchronize their states with one or two entities in the network. With the explosion of Internet-of-things and machine to machine communication, more and more devices would be using IPv6 addresses in the near future. Today, most electricity usage in United States and in developing countries are in the home buildings and commercial buildings; the electronic Internet appliances/tablets etc. are gaining popularities in the modern home networks. These network of nodes must be conscious about saving energy in order to reduce user-cost. This will eventually reduce stress on electrical grids and carbon foot-print.

IPv6 Neighbor Discovery Optimization for 6LoWPAN [6LOWPAN-ND] addresses many of the concerns described above by optimizing the Router advertisement, minimizing periodic multicast packets in the network and introducing two new options - one for node registration and another for prefix dissemination in a network where all nodes in the network are uniquely identified by their 64-bit Interface Identifier. EUI-64 identifiers are recommended as unique Interface Identifiers, however if the network is isolated from the Internet, uniqueness of the identifiers may be obtained by other mechanisms such as a random number generator with lowest collision rate. Although, the ND optimization [6LOWPAN-ND] applies to 6LoWPAN

[LOWPAN] network, the concept is mostly applicable to a power-aware IPv6 network. Therefore, this document generalizes the address registration and multicast reduction in [6LOWPAN-ND] to all IPv6 links.

Thus optimizing the regular IPv6 Neighbor Discovery [ND] to minimize total number of related signaling messages without losing generality of Neighbor Discovery and autoconfiguration and making host and router communication reliable, is desirable in any IPv6 energy-aware networks such as Home or Enterprise building networks and as well as Data Centers.

The goal of this document is to provide energy-aware and optimized Neighbor Discovery Protocols in the IPv6 subnets and links. Thus this document does not provide a solution of router advertisements and registration for 'multi-level subnets' as indicated in 6LoWPAN [LOWPAN]. In the process, the node registration method is also useful for preventing Neighbor Discovery denial of service (DOS) attacks.

The proposed changes can be used in two different ways. In one case all the hosts and routers on a link implement the new mechanisms, which gives the maximum benefits. In another case the link has a mixture of new hosts and/or routers and legacy [RFC4861] hosts and routers, operating in a mixed-mode providing some of the benefits.

In the following sections the document describes the basic operations of registration methods, optimization of Neighbor Discovery messages, interoperability with legacy IPv6 implementations and provides a section on use-case scenarios where it can be typically applicable.

2. Definition Of Terms

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

multi-level Subnets:
: It is a wireless link determined by one IPv6 off-link prefix in a network where in order to reach a destination with same prefix a packet may have to travel throguh one more 'intermediate' routers which relays the packet to the next 'intermediate' router or the host in its own.

Border Rotuer(BR):
A border router is typically located at the junction Internet and Home Network. An IPv6 router with one interface connected to IPv6 subnet and other interface connecting to a non-classic IPv6 interface such as 6LoWPAN interface. Border router is usually the gateway to the IPv6 network or Internet.

IPv6 ND-energy-aware Rotuer(NEAR):
It is the default Router of the single hop IPv6 subnet. This router implements the optimizations specified in this document. This router should be able to handle both legacy IPv6 nodes and nodes that sends registration request.

Enery-Aware Host(EAH):
A host in a IPv6 network is considered a IPv6 node without routing and forwarding capability. The EAH is the host which implements the host functionality for optimized Neighbor Discovery mentioned in this document.

Legacy IPv6 Host:
A host in a IPv6 network is considered a IPv6 node without routing and forwarding capability and implements RFC 4861 host functions.

Legacy IPv6 Router:
An IPv6 Router which implements RFC 4861 Neighbor Discovery protocols.

EUI-64:
It is the IEEE defined 64-bit extended unique identifier formed by concatenation of 24-bit or 36-bit company id value by IEEE Registration Authority and the extension identifier within that company-id assignment. The extension identifiers are 40-bit (for 24-bit company-id) or 28-bit (for the 36-bit company-id) respectively.

3. Assumptions for energy-aware Neighbor Discovery

- The energy-aware nodes in the network carry unique interface ID in the network in order to form the auto-configured IPv6 address uniquely. An EUI-64 interface ID required for global communication.
- All nodes are single IPv6-hop away from their default router in the subnet.
- /64-bit IPv6 prefix is used for Stateless Auto-address configuration (SLAAC). The IPv6 Prefix may be distributed with Router Advertisement (RA) from the default router to all the nodes in that link.

4. The set of Requirements for Energy-awareness and optimization

In future homes, machine-to-machine networks and Data-center Virtual

networks, it is essential to reduce unnecessary number of IPv6 Neighbor Discovery signalings for saving energy and saving bits in the network.

In the cloud computing environment, the concept of IPv6-subnet of link-local nodes is often extended across different networks over a Virtual LAN. Thus reducing Neighbor Discovery signaling messages is a key for enhanced services.

- o Node Registration: Node initiated Registration and address allocation is done in order to avoid periodic multicast Router Advertisement messages and often Neighbor Address resolution can be skipped as all packets go via the default router which now knows about all the registered nodes. Node Registration enables reduction of all-node and solicited-node multicast messages in the subnet.
- o Address allocation of registered nodes [ND] are performed using IPv6 Autoconfiguration [AUTOCONF].
- o Host initiated Registration and Refresh is done by sending a Router Solicitation and then a Neighbor Solicitation Messge using Address Registration Option (described below).
- o The node registration may replace the requirement of doing Duplicate Address Detection.
- o Sleepy hosts are supported by this Neighbor Discovery procedures as they are not woken up periodically by Router Advertisement multicast messages or Neighbor Solicitation multicast messages. Sleepy nodes may wake up in its own schedule and send unicast registration refresh messages when needed.
- o Since this document requires formation of an IPv6 address with an unique 64-bit Interface ID(EUI-64) is required for global IPv6 addresses. If the network is an isolated one and uses ULA [ULA] as its IPv6 address then the deployment should make sure that each MAC address in that network has unique address and can provide a unique 64-bit ID for each node in the network.
- o /64-bit Prefix is required to form the IPv6 address.
- o MTU requirement is same as IPv6 network.

5. Basic Operations

In the energy-aware IPv6 Network, the NEAR routers are the default routers for the energy-aware hosts (EAH). During the startup or joining the network the host does not wait for the Router Advertisements as the NEAR routers do not perform periodic multicast RA as per RFC 4861. Instead, the EAH sends a multicast RS to find out a NEAR router in the network. The RS message is the same as in RFC 4861. The advertising routers in the link responds to the RS message with RA with Prefix Information Option and any other options

configured in the network. If EAH hosts will look for a RA from a NEAR (E-flag) and choose a NEAR as its default router and consequently sends a unicast Neighbor Solicitation Message with ARO option in order to register itself with the default router. The EAH does not do Duplicate Address Detection or NS Resolution of addresses. NEAR maintains a binding of registered nodes and registration life-time information along with the neighbor Cache information. The NEAR is responsible for forwarding all the messages from its EAH including on-link messages from one EAH to another. For details of protocol operations please see the sections below.

When a IPv6 network consists of both legacy hosts and EAH, and if the NEAR is configured for 'mixed mode' operation, it should be able to handle ARO requests and send periodic RA. Thus it should be able to serve both energy-aware hosts and legacy hosts. Similarly, a legacy host compatible EAH falls back to RFC 4861 host behavior if a NEAR is not present in the link. See the section on 'Mixed Mode Operations' for details below.

6. Applicability Statement

This document aims to guide the implementors to choose an appropriate IPv6 neighbor discovery and Address configuration procedures suitable for any IPv6 energy-aware network. These optimization is useful for the classical IPv6 subnet and as well as future home networks, Data-Centers where saving Neighbor Discovery messages will reduce cost of control signaling and network bandwidth and as well as energy of the connected nodes. See use cases towards the end of the document.

Note that the specification allows 'Mixed-mode' operation in the energy-aware nodes for backward compatibility and transitioning to a complete energy-aware network of hosts and routers. Though the energy-aware only nodes will minimize the ND signalling and DOS attacks in the LAN.

7. New Neighbor Discovery Options and Messages

This section will discuss the registration and de-registration procedure of the hosts in the network.

7.1. Address Registration Option

The Address Registration Option(ARO) is useful for avoiding Duplicate Address Detection messages since it requires a unique ID for registration. The address registration is used for maintaining reachability of the node or host by the router. This option is

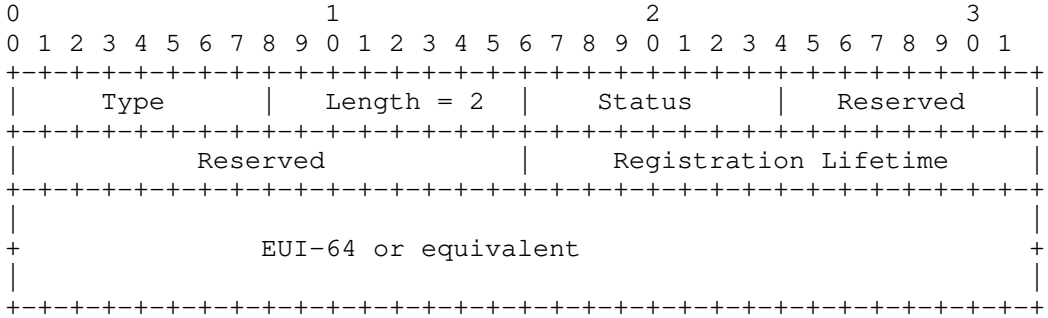
exactly the same as in [6LOWPAN-ND] which is reproduced here for the benefits of the readers.

The routers keep track of host IP addresses that are directly reachable and their corresponding link-layer addresses. This is useful for lossy and lowpower networks and as well as wired networks. An Address Registration Option (ARO) can be included in unicast Neighbor Solicitation (NS) messages sent by hosts. Thus it can be included in the unicast NS messages that a host sends as part of Neighbor Unreachability Detection to determine that it can still reach a default router. The ARO is used by the receiving router to reliably maintain its Neighbor Cache. The same option is included in corresponding Neighbor Advertisement (NA) messages with a Status field indicating the success or failure of the registration. This option is always host initiated.

The ARO is required for reliability and power saving. The lifetime field provides flexibility to the host to register an address which should be usable (the reachability information may be propagated to the routing protocols) during its intended sleep schedule of nodes that switches to frequent sleep mode.

The sender of the NS also includes the EUI-64 of the interface it is registering an address from. This is used as a unique ID for the detection of duplicate addresses. It is used to tell the difference between the same node re-registering its address and a different node (with a different EUI-64) registering an address that is already in use by someone else. The EUI-64 is also used to deliver an NA carrying an error Status code to the EUI-64 based link-local IPv6 address of the host.

When the ARO is used by hosts an SLLA option MUST be included and the address that is to be registered MUST be the IPv6 source address of the Neighbor Solicitation message.



Fields:
Type: TBD1 (See [6LOWPAN-ND])
Length: 8-bit unsigned integer. The length of the option in units of 8 bytes. Always 2.
Status: 8-bit unsigned integer. Indicates the status of a registration in the NA response. MUST be set to 0 in NS messages. See below.
Reserved: This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.
Registration Lifetime: 16-bit unsigned integer. The amount of time in a unit of 10 seconds that the router should retain the Neighbor Cache entry for the sender of the NS that includes this option.
EUI-64: 64 bits. This field is used to uniquely identify the interface of the registered address by including the EUI-64 identifier assigned to it unmodified.

The Status values used in Neighbor Advertisements are:

Status	Description
0	Success
1	Duplicate Address
2	Neighbor Cache Full
3-255	Allocated using Standards Action [RFC2434]

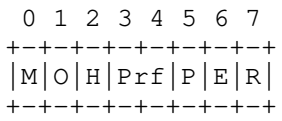
Table 1

7.2. Refresh and De-registration

A host SHOULD send a Registration messge in order to renew its registration before its registration lifetime expires in order to continue its connectivity with the network. If anytime, the node decides that it does not need the default router's service anymore, it MUST send a de-registration message - i,e, a registration message with lifetime being set to zero. A mobile host SHOULD first de-register with the default router before it moves away from the subnet.

7.3. A New Router Advertisement Flag

A new Router Advertisment flag [RF] is needed in order to distnguish a router advertisement from a energy-aware default router or a legacy IPv6 router. This flag is ignored by the legacy IPv6 hosts. EAH hosts use this flag in oder to discover a NEAR router if it receives multiple RA from both legacy and NEAR routers.



The 'E' bit above MUST be 1 when a IPv6 router implements and configures the Energy-aware Router behavior for Neighbor Discovery as per this document. All other cases E bit is 0.

The legacy IPv6 hosts will ignore the E bit in RA advertisement. All EAH MUST look for E bit in RA in order to determine the Energy-aware support in the default router in the link.

This document assumes that an implementation will have configuration knobs to determine whether it is running in classical IPv6 ND [ND] or Optimized Energy Aware ND (this document) mode or both(Mixed mode).

8. Energy-aware Neighbor Discovery Messages

Router Advertisement(RA): Periodic RAs SHOULD be avoided. Only solicited RAs are RECOMMENDED. An RA MUST contain the Source Link-layer Address option containing Router's link-layer address (this is optional in [ND]. An RA MUST carry Prefix information option with L bit being unset, so that hosts do not multicast any NS messages as part of address resolution. A new flag (E-flag) is introduced in the RA in order to characterize the energy-aware mode support. Unlike RFC4861 which suggests multicast Router Advertisements, this specification optimizes the exchange by always unicasting RAs in response to RS. This is possible since the RS always includes a SLLA option, which is used by the router to unicast the RA.

Router Solicitation(RS): Upon system startup, the node sends a multicast or link broadcast (when multicast is not supported at the link-layer) RS to find out the available routers in the link. An RS may be sent at other times as described in section 6.3.7 of RFC 4861. A Router Solicitation MUST carry Source Link-layer Address option. Since no periodic RAs are allowed in the energy-aware IPv6 network, the host may send periodic unicast RS to the

routers. The time-periods for the RS varies on the deployment scenarios and the Default Router Lifetime advertised in the RAs.

Default Router Selection: Same as in section 6.3.6 of RFC 4861[ND].

Neighbor Solicitation (NS): Neighbor solicitation is used between the hosts and the default-router as part of NUD and registering the host's address(es). An NS message MUST use the Address Registration option in order to accomplish the registration.

Neighbor Advertisement (NA): As defined in [ND] and ARO option.

Redirect Messages: A router SHOULD NOT send a Redirect message to a host since the link has non-transitive reachability. The host behavior is same as described in section 8.3 of RFC 4861[ND], i,e. a host MUST NOT send or accept redirect messages when in energy-aware mode. Same as in RFC 4861[ND]

MTU option: As per the RFC 4861.

Address Resolution: No NS/NA are sent as the prefixes are treated as off-link. Thus no address resolution is performed at the hosts. The routers keep track of Neighbor Solicitations with Address Registration options(ARO) and create an extended neighbor cache of reachable addresses. The router also knows the nexthop link-local address and corresponding link-layer address when it wants to route a packet.

Neighbor Unreachability Detection(NUD): NUD is performed in "forward-progress" fashion as described in section 7.3.1 of RFC 4861[ND]. However, if Address Registration Option is used, the NUD SHOULD be combined with the Re-registration of the node. This way no extra message for NUD is required.

9. Energy-Aware Host Behavior

A host sends Router Solicitation at the system startup and also when it suspects that one of its default routers have become unreachable(after NUD fails). The EAH MUST process the E-bit in RA as described in this document. The EAH MUST use ARO option to register with the neighboring NEAR router.

A host SHOULD be able to autoconfigure its IPv6 addresses using the IPv6 prefix obtained from Router Advertisement. The host SHOULD form

its link-local address from the EUI-64 as specified by IEEE Registration Authority and RFC 2373. If this draft feature is implemented and configured, the host MUST NOT re-direct Neighbor Discovery messages. The host does not require to join solicited-node multicast address but it MUST join the all-node multicast address.

A host always sends packets to (one of) its default router(s). This is accomplished by the routers never setting the 'L' flag in the Prefix options.

The host is unable to forward routes or participate in a routing protocol. A legacy IPv6 Host compliant EAH SHOULD be able to fall back to RFC 4861 host behavior if there is no energy-aware router (NEAR) in the link.

The energy-aware host MUST NOT send or accept re-direct messages. It does not join solicited node multicast address.

10. The Energy Aware Default Router (NEAR) Behavior

The main purpose of the default router in the context of this document is to receive and process the registration request, forward packets from one neighbor to the other, informs the routing protocol about the un-availability of the registered nodes if the routing protocol requires this information for the purpose of mobility or fast convergence. A default router (NEAR) behavior may be observed in one or more interfaces of a Border Router(BR).

A Border Router normally may have multiple interfaces and connects the nodes in a link like a regular IPv6 subnet(s) or acts as a gateway between separate networks such as Internet and home networks . The Border Router is responsible for distributing one or more /64 prefixes to the nodes to identify a packet belonging to the particular network. One or more of the interfaces of the Border Router may be connected with the energy-aware hosts or a energy-aware router(NEAR).

The Energy-Aware default router MUST not send periodic RA unless it is configured to support both legacy IPv6 and energy-aware hosts. If the Router is configured for Energy-Aware hosts support, it MUST send Router Advertisments with E-bit flag ON and MUST NOT set 'L' bit in the advertisements.

The router SHOULD NOT garbage collect Registered Neighbor Cache entries since they need to retain them until the Registration Lifetime expires. If a NEAR receives a NS message from the same host one with ARO and another without ARO then the NS message with ARO

gets the precedence and the NS without ARO is ignored. This behavior protects the router from Denial Of Service attacks. Similarly, if Neighbor Unreachability Detection on the router determines that the host is UNREACHABLE (based on the logic in [ND]), the Neighbor Cache entry SHOULD NOT be deleted but be retained until the Registration Lifetime expires. If an ARO arrives for an NCE that is in UNCREACHABLE state, that NCE should be marked as STALE.

A default router keeps a cache for all the nodes' IP addresses, created from the Address Registration processing.

10.1. Router Configuration Modes

An energy-aware Router(NEAR) MUST be able to configure in energy-aware-only mode where it will expect all hosts register with the router following RS; thus will not support legacy hosts. However, it will create legacy NCE for NS messages for other routers in the network. This mode is able to prevent ND flooding on the link.

An energy-aware Router(NEAR) SHOULD be able to have configuration knob to configure itself in Mixed-Mode where it will support both energy-aware hosts and legacy hosts. However even in mixed-mode the router should check for duplicate entries in the NCE before creating a new ones and it should rate-limit creating new NCE based on requests from the same host MAC address.

The RECOMMENDED default mode of operation for the energy-aware router is Mixed-mode.

11. NCE Management in Energy-Aware Routers

The use of explicit registrations with lifetimes plus the desire to not multicast Neighbor Solicitation messages for hosts imply that we manage the Neighbor Cache entries slightly differently than in [ND]. This results in two different types of NCEs and the types specify how those entries can be removed:

Legacy: Entries that are subject to the normal rules in [ND] that allow for garbage collection when low on memory. Legacy entries are created only when there is no duplicate NCE. In mixed-mode and energy-aware mode the legacy entries are converted to the registered entries upon successful processing of ARO. Legacy type can be considered as union of garbage-collectible and Tentative Type NCEs described in [6LOWPAN-ND].

Registered: Entries that have an explicit registered lifetime and are kept until this lifetime expires or they are explicitly unregistered.

Note that the type of the NCE is orthogonal to the states specified in [ND].

When a host interacts with a router by sending Router Solicitations that does not match with the existing NCE entry of any type, a Legacy NCE is first created. Once a node successfully registers with a Router the result is a Registered NCE. As Routers send RAs to legacy hosts, or receive multicast NS messages from other Routers the result is Legacy NCEs. There can only be one kind of NCE for an IP address at a time.

A Router Solicitation might be received from a host that has not yet registered its address with the router or from a legacy[ND] host in the Mixed-mode of operation.

In the 'Enrgy-aware' only mode the router MUST NOT modify an existing Neighbor Cache entry based on the SLLA option from the Router Solicitation. Thus, a router SHOULD create a tentative Legacy Neighbor Cache entry based on SLLA option when there is no match with the existing NCE. Such a legacy Neighbor Cache entry SHOULD be timed out in TENTATIVE_LEGACY_NCE_LIFETIME seconds unless a registration converts it into a Registered NCE.

However, in 'Mixed-mode' operation, the router does not require to keep track of TENTATIVE_LEGACY_NCE_LIFETIME as it does not know if the RS request is from a legacy host or the energy-aware hosts. However, it creates the legacy type of NCE and updates it to a registered NCE if the ARO NS request arrives corresponding to the legacy NCE. Successful processing of ARO will complete the NCE creation phase.

If ARO did not result in a duplicate address being detected, and the registration life-time is non-zero, the router creates and updates the registered NCE for the IPv6 address. if the Neighbor Cache is full and new entries need to be created, then the router SHOULD respond with a NA with status field set to 2. For successful creation of NCE, the router SHOULD include a copy of ARO and send NA to the requestor with the status field 0. A TLLA(Target Link Layer) Option is not required with this NA.

Typically for energy-aware routers (NEAR), the registration life-time and EUI-64 are recorded in the Neighbor Cache Entry along with the existing information described in [ND]. The registered NCE are meant to be ready and reachable for communication and no address resolution

is required in the link. The energy-aware hosts will renew their registration to keep maintain the state of reachability of the NCE at the router. However the router may do NUD to the idle or unreachable hosts as per [ND].

11.1. Handling ND DOS Attack

IETF community has discussed possible issues with /64 DOS attacks on the ND networks when a attacker host can send thousands of packets to the router with a on-link destination address or sending RS messages to initiate a Neighbor Solicitation from the neighboring router which will create a number of INCOMPLETE NCE entries for non-existent nodes in the network resulting in table overflow and denial of service of the existing communications.

The energy-aware behavior documented in this specification avoids the ND DOS attacks by:

- Having the hosts register with the default router
- Having the hosts send their packets via the default router
- Not resolving addresses for the Routing Solicitor by mandating SLLA option along with RS
- Checking for duplicates in NCE before the registration
- Checking against the MAC-address and EUI-64 id is possible now for NCE matches
- On-link IPv6-destinations on a particular link must be registered else these packets are not resolved and extra NCEs are not created

It is recomended that Mixed-mode operation and legacy hosts SHOULD NOT be used in the IPv6 link in order to avoid the ND DOS attacks. For the general case of Mixed-mode the router does not create INCOMPLETE NCEs for the registered hosts, but it follows the [ND] steps of NCE states for legacy hosts.

12. Mixed-Mode Operations

Mixed-Mode operation discusses the protocol behavior where the IPv6 subnet is composed with legacy IPv6 Neighbor Discovery compliant nodes and energy-aware IPv6 nodes implementing this specification.

The mixed-mode model SHOULD support the following configurations in the IPv6 link:

- The legacy IPv6 hosts and energy-aware-hosts in the network and a NEAR router
- legacy IPv6 default-router and energy-aware hosts(EAH) in the link

- o one router is in mixed mode and the link contains both legacy IPv6 hosts and EAH
- o A link contains both energy-aware IPv6 router and hosts and legacy IPv6 routers and hosts and each host should be able to communicate with each other.

In mixed-mode operation, a NEAR MUST be configured for mixed-mode in order to support the legacy IPv6 hosts in the network. In mixed-mode, the NEAR MUST act as proxy for Neighbor Solicitation for DAD and Address Resolution on behalf of its registered hosts on that link. It should follow the NCE management for the EAH as described in this document and follow RFC 4861 NCE management for the legacy IPv6 hosts. Both in mixed-mode and energy-aware mode, the NEAR sets E-bit flag in the RA and does not set 'L' on-link bit.

If a NEAR receives NS message from the same host one with ARO and another without ARO then the NS message with ARO gets the precedence.

An Energy-Aware Host implementation SHOULD support falling back to legacy IPv6 node behavior when no energy-aware routers are available in the network during the startup. If the EAH was operational in energy-aware mode and it determines that the NEAR is no longer available, then it should send a RS and find an alternate default router in the link. If no energy-aware router is indicated from the RA, then the EAH SHOULD fall back into RFC 4861 behavior. On the otherhand, in the energy-aware mode EAH SHOULD ignore multicast Router Advertisements(RA) sent by the legacy and Mixed-mode routers in the link.

The routers that are running on energy-aware mode or legacy mode SHOULD NOT dynamically switch the mode without flushing the Neighbor Cache Entries.

13. Bootstrapping

If the network is a energy-aware IPv6 subnet, and the energy-aware Neighbor Discovery mechansim is used by the hosts and routers as described in this document. At the start, the node uses its link-local address to send Router Solicitation and then it sends the Node Registration message as described in this document in order to form the address. The Duplicate address detection process should be skipped if the network is guaranteed to have unique interface identifiers which is used to form the IPv6 address.

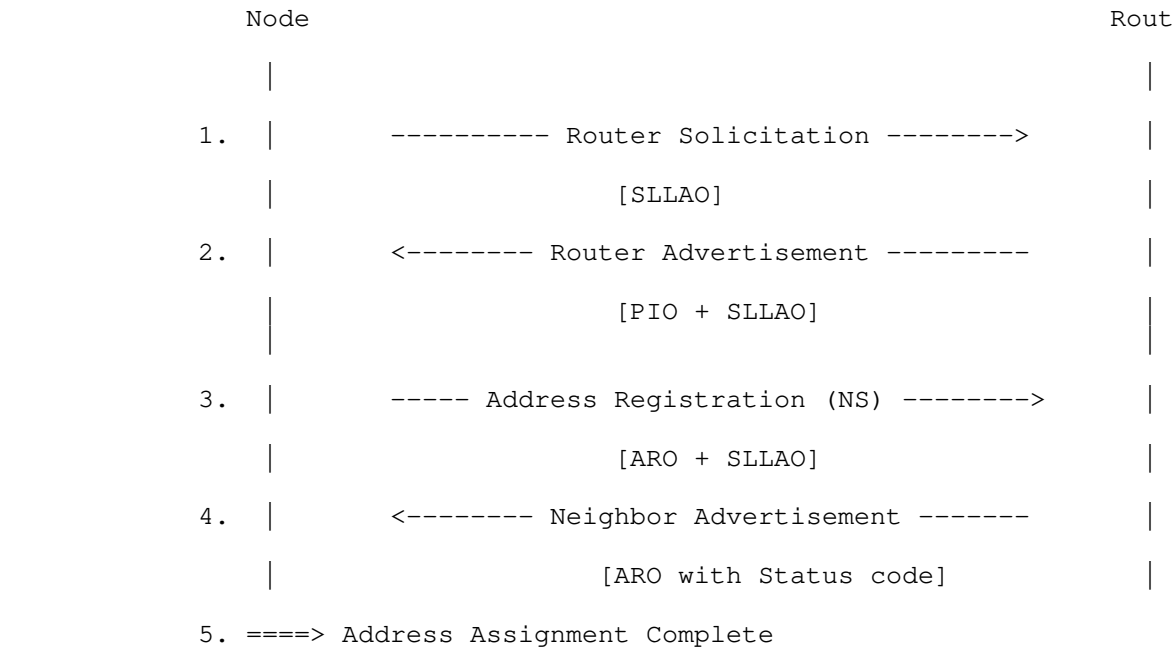


Figure 1: Neighbor Discovery Address Registration and bootstrapping

In the mixed mode operation, it is expected that logically there will be at least one legacy IPv6 router and another NEAR router present in the link. The legacy IPv6 router will follow RFC 4861 behavior and NEAR router will follow the energy-aware behavior for registration, NCE maintenance, forwarding packets from a EAH and it will also act as a ND proxy for the legacy IPv6 hosts querying to resolve a EAH node.

A legacy IPv6 host and EAH are not expected to see a difference in their bootstrapping if both legacy and energy-aware functionalities of rotuers are available in the network. It is RECOMMEDED that the EAH implementation SHOULD be able to behave like a legacy IPv6 host if it discovers that no energy-aware routing support is present in the link.

14. Handling Sleepy Nodes

The solution allows the sleepy nodes to complete its sleep schedule without waking up due to periodic Router Advertisement messages or due to Multicast Neighbor Solicitation for address resolution. The node registration lifetime SHOULD be synchronized with its sleep

interval period in order to avoid waking up in the middle of sleep for registration refresh. Depending on the application, the registration lifetime SHOULD be equal to or integral multiple of a node's sleep interval period.

15. Use Case Analysis

This section provides applicability scenarios where the energy-aware Neighbor Discovery will be most beneficial.

15.1. Data Center Routers on the link

Energy-aware Routers and hosts are useful in IPv6 networks in the Data Center as they produce less signaling and also provides ways to minimize the ND flood of messages. Moreover, this mechanism will work with data-center nodes which are deliberately in sleep mode for saving energy.

This solution will work well in Data Center Virtual network and VM scenarios where number of VLANs are very high and ND signalings are undesirably high due the multicast messaging and periodic Router Advertisments and Neighbor Unreachability detections.

15.2. Edge Routers and Home Networks

An Edge Router in the network will also benefit implementing the energy-aware Neighbor Discovery behavior in order to save the signaling and keeping track of the registered nodes in its domain. A BNG sits at the operator's edge network and often the BNG has to handle a large number of home CPEs. If a BNG runs Neighbor Discovery protocol and acts as the default router for the CPE at home, this solution will be helpful for reducing the control messages and improving network performances.

The same solution can be run on CPE or Home Residential Gateways to assign IPv6 addresses to the wired and wireless home devices without the problem of ND flooding issues and consuming less power. It provides mechanism for the sleepy nodes to adjust their registration lifetime according to their sleep schedules.

15.3. M2M Networks

Any Machine-to-machine(M2M) networks such as IPv6 surveilance networks, wireless monitoring networks and other m2m networks desire for energy-aware control protocols and dynamic address allocation. The in-built address allocation and autoconfiguration mechanism in IPv6 along with the default router capability will be useful for the

simple small-scale networks without having the burden of DHCPv6 service and Routing Protocols.

16. Mobility Considerations

If the hosts move from one subnet to another, they MUST first de-register and then register themselves in the new subnet or network. Otherwise, the regular IPv6 Mobility [IPV6M]behavior applies.

17. Updated Neighbor Discovery Constants

This section discusses the updated default values of ND constants based on [ND] section 10. New and changed constants are listed only for energy-aware-nd implementation.

Router Constants:

MAX_RTR_ADVERTISEMENTS(NEW)	3 transmissions
MIN_DELAY_BETWEEN_RAS(CHANGED)	1 second
TENTATIVE_LEGACY_NCE_LIFETIME(NEW)	30 seconds

Host Constants:

MAX_RTR_SOLICITATION_INTERVAL(NEW)	60 seconds

18. Security Considerations

These optimizations are not known to introduce any new threats against Neighbor Discovery beyond what is already documented for IPv6 [RFC 3756].

Section 11.2 of [ND] applies to this document as well.

This mechanism minimizes the possibility of ND /64 DOS attacks in energy-aware mode. See Section 11.1.

19. IANA Considerations

A new flag (E-bit) in RA has been introduced. IANA assignment of the E-bit flag is required upon approval of this document.

20. Changelog

Changes from 00 to 01:

- o Removed ABRO options and Multi-level subnet concept
- o Removed intermediate-router concept, behavior and definition
- o Added use-cases, Support for Mixed-mode operations and a diagram for bootstrapping scenario.
- o Added updates to ND constant values
- o A new co-author has beed added
- o Text for NCE Management and ND-DOS handling has been added
- o A new Router Advertisement flag has been added

21. Acknowledgements

The primary idea of this document are from 6LoWPAN Neighbor Discovery document [6LOWPAN-ND] and the discussions from the 6lowpan working group members, chairs Carsten Bormann and Geoff Mulligan and through our discussions with Zach Shelby, editor of the [6LOWPAN-ND].

The inspiration of such a IPv6 generic document came from Margaret Wasserman who saw a need for such a document at the IOT workshop at Prague IETF.

22. References

22.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[6LOWPAN-ND] Shelby, Z., Chakrabarti, S., and E. Nordmark, "ND Optimizations for 6LoWPAN", draft-ietf-6lowpan-nd-17.txt (work in progress), June 2011.

[ND] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6", RFC 4861, September 2007.

[LOWPAN] Montenegro, G. and N. Kushalnagar, "Transmission of IPv6 Packets over IEEE 802.15.4 networks", RFC 4944, September 2007.

[LOWPANG] Kushalnagar, N. and G. Montenegro, "6LoWPAN: Overview, Assumptions, Problem Statement and Goals", RFC 4919,

August 2007.

22.2. Informative References

[IPV6] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6), Specification", RFC 2460, December 1998.

[AUTOCONF] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Autoconfiguration", RFC 4862, September 2007.

[SEND] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "Secure Neighbor Discovery", RFC 3971, March 2005.

[AUTOADHOC] Baccelli, E. and M. Townsley, "IP Addressing Model in Adhoc Networks", draft-ietf-autoconf-adhoc-addr-model-02.txt (work in progress), December 2009.

[IEEE] IEEE Computer Society, "IEEE Std. 802.15.4-2003", , October 2003.

[PD] Miwakawya, S., "Requirements for Prefix Delegation", RFC 3769, June 2004.

[RF] Haberman, B. and B. Hinden, "IPv6 Router Advertisement Flags option", RFC 5175, March 2008.

[ULA] "Unique Local IPv6 Addresses", RFC 4193.

[IPV6M] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.

Authors' Addresses

Samita Chakrabarti
Ericsson
San Jose, CA
USA

Email: samita.chakrabarti@ericsson.com

Erik Nordmark
Cisco Systems
San Jose, CA
USA

Email: nordmark@cisco.com

Margaret Wasserman
Painless Security

Email: mrw@lilacglade.org

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: May 3, 2012

T. Narten, Ed.
IBM
M. Sridharan
Microsoft
D. Dutt
Cisco
D. Black
EMC
L. Kreeger
Cisco

October 31, 2011

Problem Statement: Overlays for Network Virtualization
draft-narten-nvo3-overlay-problem-statement-01

Abstract

This document describes issues associated with providing multi-tenancy in large data center networks and an overlay-based network virtualization approach to addressing them. A key multi-tenancy requirement is traffic isolation, so that a tenant's traffic is not visible to any other tenant. This isolation can be achieved by assigning one or more virtual networks to each tenant such that traffic within a virtual network is isolated from traffic in other virtual networks. The primary functionality required is provisioning virtual networks, associating a virtual machine's NIC with the appropriate virtual network, and maintaining that association as the virtual machine is activated, migrated and/or deactivated. Use of an overlay-based approach enables scalable deployment on large network infrastructures.

Table of Contents

1. Introduction

Server virtualization is increasingly becoming the norm in data centers. With server virtualization, each physical server supports multiple virtual machines (VMs), each running its own operating system, middleware and applications. Virtualization is a key enabler of workload agility, i.e., allowing any server to host any application and providing the flexibility of adding, shrinking, or moving services within the physical infrastructure. Server virtualization provides numerous benefits, including higher utilization, increased data security, reduced user downtime, reduced power usage, etc.

Large scale multi-tenant data centers are taking advantage of the benefits of server virtualization to provide a new kind of hosting, a virtual hosted data center. Multi-tenant data centers are ones in which each tenant could belong to a different company (in the case of a public provider) or a different department (in the case of a internal company data center). Each tenant has the expectation of a level of security and privacy separating their resources from those of other tenants. Each virtual data center looks similar to its physical counterpart, consisting of end stations connected by a network, complete with services such as load balancers and firewalls. The network within each virtual data center can be a pure routed network, a pure bridged network or a combination of bridged and routed network. The key requirement is that each such virtual network is isolated from the others, whether the networks belong to the same tenant or different tenants.

This document outlines the problems encountered in scaling the number of isolated networks in a data center, as well as the problems of managing the creation/deletion, membership and span of these networks and makes the case that an overlay based approach, where individual networks are implemented within individual virtual networks that are dynamically controlled by a standardized control plane provides a number of advantages over current approaches. The purpose of this document is to identify the set of problems that any solution has to address in building multi-tenant data centers. With this approach, the goal is to allow the construction of standardized, interoperable implementations to allow the construction of multi-tenant data centers.

Section 2 describes the problem space details. Section 3 defines virtual networks. Section 4 provides a general discussion of overlays and standardization issues. Section 5 discusses the control plane issues that require addressing for virtual networks. Section 6 and 7 discuss related work and further work.

2. Problem Details

The following subsections describe aspects of multi-tenant networking that pose problems for large scale network infrastructure. Different problem aspects may arise based on the network architecture and scale.

2.1. Multi-tenant Environment Scale

Cloud computing involves on-demand elastic provisioning of resources for multi-tenant environments. A common example of cloud computing is the public cloud, where a cloud service provider offers these elastic services to multiple customers over the same infrastructure. This elastic on-demand nature in conjunction with trusted hypervisors to control network access by VMs calls for resilient distributed network control mechanisms.

2.2. Virtual Machine Mobility Requirements

A key benefit of server virtualization is virtual machine (VM) mobility. A VM can be migrated from one server to another, live i.e. as it continues to run and without shutting down the VM and restarting it at a new location. A key requirement for live migration is that a VM retain its IP address(es) and MAC address(es) in its new location (to avoid tearing down existing communication). Today, servers are assigned IP addresses based on their physical location, typically based on the ToR (Top of Rack) switch for the server rack or the VLAN configured to the server. This works well for physical servers, which cannot move, but it restricts the placement and movement of the more mobile VMs within the data center (DC). Any solution for a scalable multi-tenant DC must allow a VM to be placed (or moved to) anywhere within the data center, without being constrained by the subnet boundary concerns of the host servers.

2.3. Span of Virtual Networks

Another use case is cross pod expansion. A pod typically consists of one or more racks of servers with its associated network and storage connectivity. Tenants may start off on a pod and, due to expansion, require servers/VMs on other pods, especially the case when tenants on the other pods are not fully utilizing all their resources. This use case requires that virtual networks span multiple pods in order to provide connectivity to all of the tenant’s servers/VMs.

2.4. Inadequate Forwarding Table Sizes in Switches

Today's virtualized environments place additional demands on the forwarding tables of switches. Instead of just one link-layer address per server, the switching infrastructure has to learn addresses of the individual VMs (which could range in the 100s per server). This is a requirement since traffic from/to the VMs to the rest of the physical network will traverse the physical network infrastructure. This places a much larger demand on the switches' forwarding table capacity compared to non-virtualized environments, causing more traffic to be flooded or dropped when the addresses in use exceeds the forwarding table capacity.

2.5. Decoupling Logical and Physical Configuration

Data center operators must be able to achieve high utilization of server and network capacity. For efficient and flexible allocation, operators should be able to spread a virtual network instance across servers in any rack in the data center. It should also be possible to migrate compute workloads to any server anywhere in the network while retaining the workload's addresses. This can be achieved today by stretching VLANs (e.g., by using TRILL or OTV).

However, in order to limit the broadcast domain of each VLAN, multi-destination frames within a VLAN should optimally flow only to those devices that have that VLAN configured. When workloads migrate, the physical network (e.g., access lists) may need to be reconfigured which is typically time consuming and error prone.

2.6. Support Communication Between VMs and Non-virtualized Devices

Within data centers, not all communication will be between VMs. Network operators will continue to use non-virtualized servers for various reasons, traditional routers to provide L2VPN and L3VPN services, traditional load balancers, firewalls, intrusion detection engines and so on. Any virtual network solution should be capable of working with these existing systems.

2.7. Overlay Design Characteristics

There are existing layer 2 overlay protocols in existence, but they were not necessarily designed to solve the problem in the environment of a highly virtualized data center. Below are some of the characteristics of environments that must be taken into account by the overlay technology:

1. Highly distributed systems. The overlay should work in an environment where there could be many thousands of access switches (e.g. residing within the hypervisors) and many more end systems (e.g. VMs) connected to them. This leads to a distributed mapping system that puts a low overhead on the overlay tunnel endpoints.

2. Many highly distributed virtual networks with sparse connectivity. Each virtual network could be highly dispersed inside the data center. Also, along with expectation of many virtual networks, the number of end systems connected to any one virtual network is expected to be relatively low; Therefore, the percentage of access switches participating in any given virtual network would also be expected to be low. For this reason, efficient pruning of multi-destination traffic should be taken into consideration.

3. Highly dynamic end systems. End systems connected to virtual networks can be very dynamic, both in terms of creation/deletion/ power-on/off and in terms of mobility across the access switches.

4. Work with existing, widely deployed network Ethernet switches and IP routers without requiring wholesale replacement. The first hop switch that adds and removes the overlay header will require new equipment and/or new software.

5. Network infrastructure administered by a single administrative domain. This is consistent with operation within a data center, and not across the Internet.

3. Defining Virtual Networks and Tenants

Virtual Networks are used to isolate a tenant's traffic from other tenants (or even traffic within the same tenant that requires isolation). There are two main characteristics of virtual networks:

1. Providing network address space that is isolated from other virtual networks. The same network addresses may be used in different virtual networks on the same underlying network infrastructure.

2. Limiting the scope of frames to not exit a virtual network except through controlled exit points or "gateways".

3.1. Limitations of Existing Virtual Network Models

Virtual networks are not new to networking. VLANs are a well known construct in the networking industry. VLAN is a bridging construct which provides the semantics of virtual networks mentioned above: a MAC address is unique within a VLAN, but not necessarily across VLANs and broadcast traffic is limited to the VLAN it originates from. In the case of IP networks, routers have the concept of a Virtual Routing and Forwarding (VRF). The same router can run multiple instances of routing protocols, each with their own forwarding table. Each instance is referred to as a VRF, which is a mechanism that provides address isolation. Since broadcasts are never forwarded across IP subnets, limiting broadcasts are not applicable to VRFs. In the case of both VLAN and VRF, the forwarding table is looked up using the tuple {VLAN, MAC address} or {VRF, IP address}.

But there are two problems with these constructs. VLANs are a pure bridging construct while VRF is a pure routing construct. VLANs are carried along with a frame to allow each forwarding point to know what VLAN the frame belongs to. VLAN today is defined as a 12 bit number, limiting the total number of VLANs to 4096 (though typically, this number is 4094 since 0 and 4095 are reserved). Due to the large number of tenants that a cloud provider might service, the 4094 VLAN limit is often inadequate. In addition, there is often a need for multiple VLANs per tenant, which exacerbates the issue.

There is no VRF indicator carried in frames. The VRF is derived at each hop using a combination of incoming interface and some information in the frame. Furthermore, the VRF model has typically assumed that a separate control plane governs the population of the forwarding table within that VRF. Thus, a traditional VRF model assumes multiple, independent control planes and has no specific tag within a frame to identify the VRF of the frame.

3.2. Virtual Network Instance

To overcome the limitations of a traditional VLAN or VRF model, we define a new mechanism for virtual networks called a virtual network instance. Each virtual network is assigned a virtual network instance ID, shortened to VNID for convenience. A virtual network instance provides the semantics of a virtual network: address disambiguation and multi-destination frame scoping. A virtual network can be either routed or bridged. So, a VNID can be used for both bridged networks and routed networks and so is unlike a VLAN or a VRF. To build large multi-tenant data centers, a larger number space than the 12b VLAN is required. 24 bits is the most common value identified by multiple solutions that attempt to address this problem space (or similar problem spaces). To simplify the building and

administration of these large data centers, we require that the VNID be carried with each frame (similar to a VLAN, but unlike a VRF). Finally, because of the nature of a virtual data center and to allow scaling virtual networks to massive scales, we don't require a separate control plane to run for each virtual network. We'll identify other possible mechanisms to populate the forwarding tables for virtual networks in section 5.1.

3.3. Tenant

Tenant is the administrative entity that that is responsible for and manages a specific virtual network and its associated services (whether virtual or physical). In a cloud environment, a tenant would correspond to the customer that has defined and is using a particular virtual network. However, there is a one-to-many mapping between tenants and virtual network instances. A single tenant may operate multiple individual virtual networks, each associated with a different service.

4. Network Overlays

To address the problems of decoupling physical and logical configuration and allowing VM mobility without exploding the forwarding table sizes in the switches and routers, a network overlay model can be used.

The idea behind an overlay is quite straightforward. The original frame is encapsulated by the first hop network device. The encapsulation identifies the destination as the device that will perform the decapsulation before delivering the frame to the endpoint. The rest of the network forwards the frame based on the encapsulation header and can be oblivious to the payload that is carried inside. To avoid belaboring the point each time, the first hop network device can be a traditional switch or router or the virtual switch residing inside a hypervisor. Furthermore, the endpoint can be a VM or it can be a physical server. Some examples of network overlays are tunnels such as IP GRE [RFC2784], LISP[I-D.ietf-lisp] or TRILL [RFC6325].

With an overlay, the VNID can be carried within the overlay header so that every frame has its VNID explicitly identified in the frame. Since both routed and bridged semantics can be supported by a virtual data center, the original frame carried within the overlay header can be an Ethernet frame complete with MAC addresses or just the IP packet.

4.1. Benefits of an Overlay Approach

The use of a large (e.g., 24-bit) VNID would allow 16 million distinct virtual networks within a single data center, eliminating current VLAN size limitations. This VNID needs to be carried in the data plane along with the packet. Adding an overlay header provides a place to carry this VNID.

A key aspect of overlays is the decoupling of the "virtual" MAC and IP addresses used by VMs from the physical network infrastructure and the infrastructure IP addresses used by the data center. If a VM changes location, the switches at the edge of the overlay simply update their mapping tables to reflect the new location of the VM within the data center's infrastructure space. Because an overlay network is used, a VM can now be located anywhere in the data center that the overlay reaches without regards to traditional constraints implied by L2 properties such as VLAN numbering, or the span of an L2 broadcast domain scoped to a single pod or access switch.

Multi-tenancy is supported by isolating the traffic of one virtual network instance from traffic of another. Traffic from one virtual network instance cannot be delivered to another instance without (conceptually) exiting the instance and entering the other instance via an entity that has connectivity to both virtual network instances. Without the existence of this entity, tenant traffic remains isolated within each individual virtual network instance. External communications (from a VM within a virtual network instance to a machine outside of any virtual network instance, e.g. on the Internet) is handled by having an ingress switch forward traffic to an external router, where an egress switch decapsulates a tunneled packet and delivers it to the router for normal processing. This router is external to the overlay, and behaves much like existing external facing routers in data centers today.

Overlays are designed to allow a set of VMs to be placed within a single virtual network instance, whether that virtual network provides the bridged network or a routed network.

4.2. Standardization Issues for Overlay Networks

4.2.1. Overlay Header Format

Different overlay header formats are possible as are different possible encodings of the VNID. Existing overlay headers maybe extended or new ones defined. This document does not address the exact header format or VNID encoding except to state that any solution MUST:

1. Carry the VNID in each frame

2. Allow the payload to be either a complete Ethernet frame or only an IP packet

4.2.2. Fragmentation

Whenever tunneling is used, one faces the potential problem that the packet plus the encapsulation overhead will exceed the MTU of the path to the egress router. If the outer encapsulation is IP, fragmentation could be left to the IP layer, or it could be done at the overlay level in a more optimized fashion that is independent of the overlay encapsulation header, or it could be left out altogether, if it is believed that data center networks can be engineered to prevent MTU issues from arising.

Related to fragmentation is the question of how best to handle Path MTU issues, should they occur. Ideally, the original source of any packet (i.e, the sending VM) would be notified of the optimal MTU to use. Path MTU problems occurring within an overlay network would result in ICMP MTU exceeded messages being sent back to the egress tunnel switch at the entry point of the overlay. If the switch is embedded within a hypervisor, the hypervisor could notify the VM of a more appropriate MTU to use. It may be appropriate to specify a set of best practices for implementers related to the handling of Path MTU issues.

4.2.3. Checksums and FCS

When tunneling packets, both the inner and outer headers could have their own checksum, duplicating effort and impacting performance. Therefore, we strongly recommend that any solution carry only one set of checksum or frame FCS.

When the inner packet is TCP or UDP, they already include their own checksum, and adding a second outer checksum (using the same 1's complement algorithm) provides little value. Similarly, if the inner packet is an Ethernet frame, the frame FCS protects the original frame and a new frame FCS over both the original frame and the overlay header protects the new encapsulated frame.

In IPv4, UDP checksums can be disabled on a per-packet basis simply by setting the checksum field to zero. IPv6, however, specifies that UDP checksums must always be included. But even for IPv6, the LISP protocol[I-D.ietf-lisp] already allows a zero checksum field. The 6man working group is also currently considering relaxing the IPv6 UDP checksum requirement [I-D.ietf-6man-udpzero].

For Ethernet frames, L2 overlays such as TRILL already mandate only a single frame FCS.

4.2.4. Middlebox Traversal

One issue to consider is to whether the overlay will need to run over networks that include middleboxes such as NAT. Middleboxes may have difficulty properly supporting multicast or other aspects of an overlay header. Inside a data center, it may well be the case that middlebox traversal is a non-issue. But if overlays are extended across the broader Internet, the presence of middleboxes may be of concern.

4.2.5. OAM

Successful deployment of an overlay approach will likely require appropriate Operations, Administration and Maintenance (OAM) facilities.

5. Control Plane

The control plane needs to address the following pieces, at least:

1. A mechanism to populate the forwarding table of a virtual network instance.

2. A mechanism to handle multi-destination frames within a virtual network instance.

3. A mechanism to allow an endpoint to inform the access switch which virtual network instance it wishes to join on a virtual network interface.

4. A mechanism to allow an endpoint to inform the access switch about its leaving the network so that the access switch can clean up state.

5.1. Populating the Forwarding Table of a Virtual Network Instance

When an access switch has to forward a frame from one endpoint to another, across the network, it has to consult some form of a forwarding table. When we use network overlays, the problem boils down to deriving the mapping between the inner and outer addresses i.e. deriving the destination address in the overlay header based on the destination address sent by the endpoint. Two well known mechanisms for populating the forwarding table (or deriving the mapping table) of a switch are (i) via a routing control protocol and

(ii) learning from the data plane as Ethernet bridges do. Another mechanism is through a centralized mapping database. Any solution must avoid problems associated with scaling a virtual network instance across a large data center.

5.2. Handling Multi-destination Frames

Another aspect of address mapping concerns the handling of multi-destination frames, i.e. broadcast and multicast frames, or the delivery of unicast packets when no mapping exists. Associating a infrastructure multicast address is one possible way of connecting together all the machines belonging to the same VNID. However, existing multicast implementations do not scale to efficiently handle hundreds of thousands of multicast groups, as would be required if one multicast group were assigned to each VNID.

5.3. Associating a VNID With An Endpoint

When an endpoint, such as VM or physical server, connects to the infrastructure, we must define a mechanism to allow the endpoint to identify to the access switch the network instance that it wishes to join. Typically, it is a virtual NIC (the one connected to the VM) coming up that triggers this association. The access switch can then determine the VNID to be associated with this virtual NIC. A standard protocol that all types of overlay encapsulation points can use to identify the VNID associated with an endpoint will be beneficial for supporting multi-vendor implementations. This protocol could also be used to distribute any per virtual network information (e.g. a multicast group address). This signaling can provide the stimulus to trigger the overlay termination points to perform any actions needed within the infrastructure network (e.g. use IGMP to join a multicast group).

5.4. Disassociating a VNID on Termination or Move

To enable cleaning up state in the access switch, we must define a mechanism to allow an endpoint to signal its disconnection from the network.

6. Related Work

6.1. ARMD

ARMD is chartered to look at data center scaling issues with a focus on address resolution. ARMD is currently chartered to develop a problem statement and is not currently developing solutions. While an overlay-based approach may address some of the "pain points" that

have been raised in ARMD (e.g., better support for multi-tenancy), an overlay approach may also push some of the L2 scaling concerns (e.g., excessive flooding) to the IP level (flooding via IP multicast). Analysis will be needed to understand the scaling trade offs of an overlay based approach compared with existing approaches. On the other hand, existing IP-based approaches such as proxy ARP may help mitigate some concerns.

6.2. TRILL

TRILL is an L2 based approach aimed at improving deficiencies and limitations with current Ethernet networks. Approaches to extend TRILL to support more than 4094 VLANs are currently under investigation [I-D.eastlake-trill-rbridge-fine-labeling]

6.3. L2VPNs

The IETF has specified a number of approaches for connecting L2 domains together as part of the L2VPN Working Group. That group, however has historically been focused on Provider-provisioned L2 VPNs, where the service provider participates in management and provisioning of the VPN. In addition, much of the target environment for such deployments involves carrying L2 traffic over WANs. Overlay approaches are intended be used within data centers where the overlay network is managed by the data center operator, rather than by an outside party. While overlays can run across the Internet as well, they will extend well into the data center itself (e.g., up to and including hypervisors) and include large numbers of machines within the data center itself.

Other L2VPN approaches, such as L2TP [RFC2661] require significant tunnel state at the encapsulating and decapsulating end points. Overlays require less tunnel state than other approaches, which is important to allow overlays to scale to hundreds of thousands of end points. It is assumed that smaller switches (i.e., virtual switches in hypervisors or the physical switches to which VMs connect) will be part of the overlay network and be responsible for encapsulating and decapsulating packets.

6.4. Proxy Mobile IP

Proxy Mobile IP [RFC5213] [RFC5844] makes use of the GRE Key Field [RFC5845] [RFC6245], but not in a way that supports multi-tenancy.

6.5. LISP

LISP[I-D.ietf-lisp] essentially provides an IP over IP overlay where the internal addresses are end station Identifiers and the outer IP

addresses represent the location of the end station within the core IP network topology. The LISP overlay header uses a 24 bit Instance ID used to support overlapping inner IP addresses.

6.6. Individual Submissions

Many individual submissions also look to addressing some or all of the issues addressed in this draft. Examples of such drafts are VXLAN [I-D.mahalingam-dutt-dcops-vxlan], NVGRE [I-D.sridharan-virtualization-nvgre] and Virtual Machine Mobility in L3 networks[I-D.wkumari-dcops-l3-vmmobility].

7. Further Work

It is believed that overlay-based approaches may be able to reduce the overall amount of flooding and other multicast and broadcast related traffic (e.g, ARP and ND) currently experienced within current data centers with a large flat L2 network. Further analysis is needed to characterize expected improvements.

8. Summary

This document has argued that network virtualization using L3 overlays addresses a number of issues being faced as data centers scale in size. In addition, careful consideration of a number of issues would lead to the development of interoperable implementation of virtualization overlays.

9. Acknowledgments

Helpful comments and improvements to this document have come from Ariel Hendel, Vinit Jain, and Benson Schliesser.

10. IANA Considerations

This memo includes no request to IANA.

11. Security Considerations

TBD

12. Informative References

[I-D.eastlake-trill-rbridge-fine-labeling]
Eastlake, D., Zhang, M., Agarwal, P., Dutt, D., and R. Perlman, "RBridges: Fine-Grained Labeling", draft-eastlake-trill-rbridge-fine-labeling-02 (work in progress), October 2011.

[I-D.hasmit-otv]
Grover, H., Rao, D., Farinacci, D., and V. Moreno, "Overlay Transport Virtualization", draft-hasmit-otv-03 (work in progress), July 2011.

[I-D.ietf-6man-udpzero]
Fairhurst, G. and M. Westerlund, "IPv6 UDP Checksum Considerations", draft-ietf-6man-udpzero-04 (work in progress), October 2011.

[I-D.ietf-lisp]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-15 (work in progress), July 2011.

[I-D.mahalingam-dutt-dcops-vxlan]
Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-00 (work in progress), August 2011.

[I-D.sridharan-virtualization-nvgre]
Sridharan, M., Duda, K., Ganga, I., Greenberg, A., Lin, G., Pearson, M., Thaler, P., Tumuluri, C., Venkataramaiah, N., and Y. Wang, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-00 (work in progress), September 2011.

[I-D.wkumari-dcops-l3-vmmobility]
Kumari, W. and J. Halpern, "Virtual Machine mobility in L3 Networks.", draft-wkumari-dcops-l3-vmmobility-00 (work in progress), August 2011.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P.

Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, September 2000.

[RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.

[RFC5844] Wakikawa, R. and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6", RFC 5844, May 2010.

[RFC5845] Muhanna, A., Khalil, M., Gundavelli, S., and K. Leung, "Generic Routing Encapsulation (GRE) Key Option for Proxy Mobile IPv6", RFC 5845, June 2010.

[RFC6245] Yegani, P., Leung, K., Lior, A., Chowdhury, K., and J. Navali, "Generic Routing Encapsulation (GRE) Key Extension for Mobile IPv4", RFC 6245, May 2011.

[RFC6325] Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

Authors' Addresses

Thomas Narten (editor)
IBM

Email: narten@us.ibm.com

Murari Sridharan
Microsoft

Email: muraris@microsoft.com

Dinesh Dutt
Cisco

Email: ddutt@cisco.com

David Black
EMC

Email: david.black@emc.com

Lawrence Kreeger
Cisco

Email: kreeger@cisco.com